3D object modeling and recognition from photographs and image sequences

Fred Rothganger, Svetlana Lazebnik, Cordelia Schmid, Jean Ponce

3D Object Modeling and Recognition from Photographs and Image Sequences

Fred Rothganger[1], Svetlana Lazebnik[1], Cordelia Schmid[2], and Jean Ponce[1]

[1] Department of Computer Science and Beckman Institute
University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA
{rothgang,slazebni,jponce}@uiuc.edu
[2] INRIA Rhône-Alpes
665, Avenue de l'Europe, 38330 Montbonnot, France
Cordelia.Schmid@inrialpes.fr

Abstract. This chapter proposes a representation of rigid three-dimensional (3D) objects in terms of local affine-invariant descriptors of their images and the spatial relationships between the corresponding surface patches. Geometric constraints associated with different views of the same patches under affine projection are combined with a normalized representation of their appearance to guide the matching process involved in object modeling and recognition tasks. The proposed approach is applied in two domains: (1) Photographs — models of rigid objects are constructed from small sets of images and recognized in highly cluttered shots taken from arbitrary viewpoints. (2) Video — dynamic scenes containing multiple moving objects are segmented into rigid components, and the resulting 3D models are directly matched to each other, giving a novel approach to video indexing and retrieval.

1 Introduction

Traditional feature-based geometric approaches to three-dimensional (3D) object recognition — such as alignment [13, 19] or geometric hashing [15] — enumerate various subsets of geometric image features before using pose consistency constraints to confirm or discard competing match hypotheses. They largely ignore the rich source of information contained in the image brightness and/or color pattern, and thus typically lack an effective mechanism for selecting promising matches. Appearance-based methods, as originally proposed in the context of face recognition [43] and 3D object recognition [28], prefer a classical pattern recognition framework that exploits the discriminatory power of (relatively) low-dimensional, empirical models of global object appearance in classification tasks. However, they typically de-emphasize the combinatorial aspects of the search involved in any matching task, which limits their ability to handle occlusion and clutter.

Viewpoint and/or illumination invariants provide a natural indexing mechanism for object recognition tasks. Unfortunately, although planar objects and

certain simple shapes—such as bilateral symmetries or various types of generalized cylinders—admit invariants, general 3D shapes do not [4], which is the main reason why invariants have fallen out of favor after an intense flurry of activity in the early 1990s [26, 27]. In this chapter, we revisit invariants as a *local* description of truly three-dimensional objects: Indeed, although smooth surfaces are almost never planar in the large, they are always planar in the small —that is, sufficiently small patches can be treated as being comprised of coplanar points. Concretely, we propose to capture the appearance of salient surface patches using local image descriptors that are invariant under affine transformations of the spatial domain [18, 24] and of the brightness signal [20], and to capture their spatial relationships using multi-view geometric constraints related to those studied in the structure from motion literature [39]. This representation is directly related to a number of recent schemes for combining the local surface appearance at "interest points" [12] with geometric constraints in tasks such as wide-baseline stereo matching [44], image retrieval [36], and object recognition [20]. These methods normally either require storing a large number of views for each object, or limiting the range of admissible viewpoints. In contrast, our approach supports the automatic acquisition of explicit 3D object models from multiple unregistered images, and their recognition in photographs and videos taken from arbitrary viewpoints.

Section 2 presents the main elements of our object representation framework. It is applied in Sections 3 and 4 to the automated acquisition of 3D object models from small sets of unregistered images and to the identification and localization of these models in cluttered photographs taken from arbitrary and unknown viewpoints. Section 5 briefly discusses further applications to the video indexing and retrieval domain, including a method for segmenting dynamic scenes observed by a moving camera into rigid components and matching the 3D models recovered from different shots. We conclude in Section 6 with a short discussion of the promise and limitations of the proposed approach.

2 Approach

2.1 Affine Regions and their Description

The construction of local invariant models of object appearance involves two steps, the detection of salient image regions, and their description. Ideally, the regions found in two images of the same object should be the projections of the same surface patches. Therefore, they must be *covariant*, with regions detected in the first picture mapping onto those found in the second one via the geometric and photometric transformations induced by the corresponding viewpoint and illumination changes. In turn, detection must be followed by a description stage that constructs a region representation *invariant* under these changes. For small patches of smooth Lambertian surfaces, the transformations are (to first order) affine, and we use the approach recently proposed by Mikolajczyk and Schmid to find the corresponding *affine regions*: Briefly, the algorithm iterates over steps where (1) an elliptical image region is deformed to maximize the isotropy of the

corresponding brightness pattern (shape adaptation [10]); (2) its characteristic scale is determined as a local extremum of the normalized Laplacian in scale space (scale selection [17]); and (3) the Harris operator [12] is used to refine the position of the the ellipse's center (localization [24]). The scale-invariant interest point detector proposed in [23] provides an initial guess for this procedure, and the elliptical region obtained at convergence can be shown to be covariant under affine transformations. The affine region detection process used in this chapter implements both this algorithm and a variant where a difference-of-Gaussians (DoG) operator replaces the Harris interest point detector. Note that this operator tends to find corners and points where significant intensity changes occur, while the DoG detector is (in general) attracted to the centers of roughly uniform regions (blobs): Intuitively, the two operators provide complementary kinds of information (see Figure 1 for examples).

Fig. 1. Affine regions found by Harris-Laplacian (left) and DoG (right) detectors.

The affine regions output by our detection process are ellipses that can be mapped onto a unit circle centered at the origin using a one-parameter family of affine transformations. This ambiguity can be resolved by determining the dominant gradient orientation of the image region, turning the corresponding ellipse into a parallelogram and the unit circle into a square (Figure 2). Thus, the output of the detection process is a set of image regions in the shape of parallelograms, together with affine *rectifying transformations* that map each parallelogram onto a "unit" square centered at the origin (Figure 3).

A rectified affine region is a normalized representation of the *local* surface appearance. For distant observers (affine projection), it is invariant under arbitrary viewpoint changes. For Lambertian patches and distant light sources, it can also be made invariant to changes in illumination (ignoring shadows) by subtracting the mean patch intensity from each pixel value and normalizing the Frobenius norm of the corresponding image array to one. The Euclidean distance between feature vectors associated with their pixel values can thus be used to compare rectified patches, irrespective of viewpoint and (affine) illumination changes. Other feature spaces may of course be used as well. As many others, we have found the Lowe's SIFT descriptor [20] —a histogram over both spatial dimensions and gradient orientations— to perform well in our experiments, along

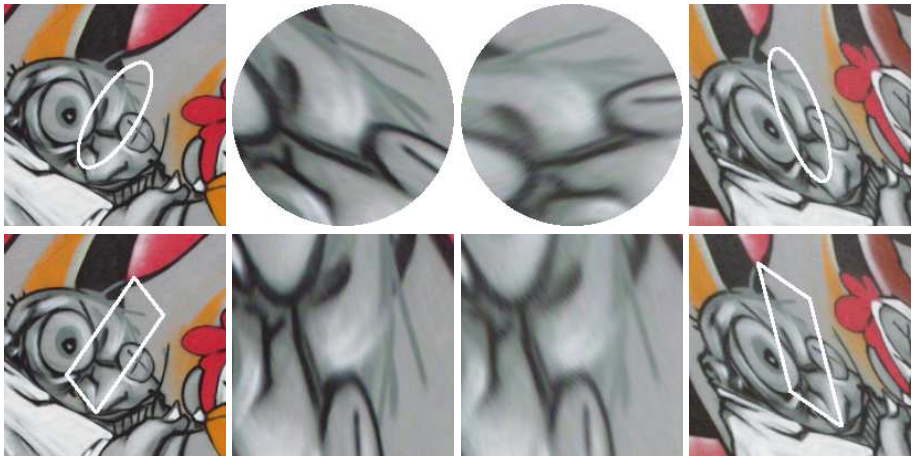


Fig. 2. Normalizing patches. The left two columns show a patch from image 1 of Krystian Mikolajczyk's graffiti dataset (available from the INRIA LEAR group's web page: `http://lear.inrialpes.fr/software`). The right two columns show the matching patch from image 4. The first row shows the ellipse determined by affine adaptation. This normalizes the shape, but leaves a rotation ambiguity, as illustrated by the normalized circles in the center. The second row shows the same patches with orientation determined by the gradient at about twice the characteristic scale.

with a 10×10 color histogram drawn from the UV portion of YUV space when color is available.

2.2 Geometric Constraints

Given an affine region, let us denote by $\mathcal{R}$ the affine transformation from the image patch to its rectified (normalized) form, and by $\mathcal{S} = \mathcal{R}^{-1}$ the affine transformation from the rectified form back to the image patch (Figure 3). The 3×3 matrix $\mathcal{S}$ has the form

$$\mathcal{S} = \begin{bmatrix} \boldsymbol{h} & \boldsymbol{v} & \boldsymbol{c} \\ 0 & 0 & 1 \end{bmatrix},$$

and its columns enjoy the following geometric interpretation: The third column gives the homogeneous coordinates of the center c of the corresponding image parallelogram, while $\boldsymbol{h}$ and $\boldsymbol{v}$ are the vectors joining c to the midpoints of the parallelogram's sides (Figure 3). The matrix $\mathcal{S}$ effectively contains the locations of three points in the image, so a match between $m \geq 2$ images of the same patch contains exactly the same information as a match between m triples of points. It is thus clear that all the machinery of structure from motion [39] and pose estimation [13, 19] from point matches can be exploited in modeling and object recognition tasks. Reasoning in terms of multi-view constraints associated with the matrix $\mathcal{S}$ provides a unified and convenient representation for all stages of both tasks.

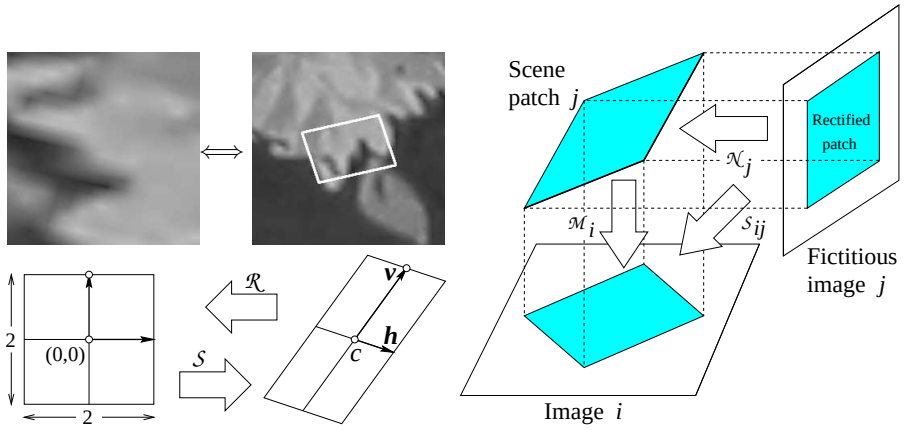


Fig. 3. Geometric structure. Top left: A rectified patch and the original image region. Bottom left: Interpretation of the rectification matrix $\mathcal{R}$ and its inverse $\mathcal{S}$. Right: Interpretation of the decomposition of the mapping $\mathcal{S}_{ij}$ into the product of a projection matrix $\mathcal{M}_i$ and an inverse projection matrix $\mathcal{N}_j$.

Suppose there are n surface patches observed in m images, and that we are given a complete set of measurements $\mathcal{S}_{ij}$ as defined above for image indices $i = 1, \ldots, m$ and patch indices $j = 1, \ldots, n$. (Later, we will show how to handle the "missing data" problem that results when not all patches are visible in all views.) A rectified patch can be thought of as a fictitious view of the original surface patch (Figure 3), and the mapping $\mathcal{S}_{ij}$ can thus be decomposed into an *inverse projection* $\mathcal{N}_j$ [5] that maps the rectified patch onto the corresponding surface patch, followed by a projection $\mathcal{M}_i$ that maps that patch onto its projection in image number i. In particular, we can write

$$\hat{\mathcal{S}} \stackrel{\text{def}}{=} \begin{bmatrix} \mathcal{S}_{11} & \ldots & \mathcal{S}_{1n} \\ \vdots & \ddots & \vdots \\ \mathcal{S}_{m1} & \ldots & \mathcal{S}_{mn} \end{bmatrix} = \begin{bmatrix} \mathcal{M}_1 \\ \vdots \\ \mathcal{M}_m \end{bmatrix} \begin{bmatrix} \mathcal{N}_1 & \ldots & \mathcal{N}_n \end{bmatrix}.$$

The inverse projection matrix can be written as

$$\mathcal{N}_j = \begin{bmatrix} \boldsymbol{H} & \boldsymbol{V} & \boldsymbol{C} \\ 0 & 0 & 1 \end{bmatrix}_j,$$

and its columns admit a geometric interpretation similar to that of $\mathcal{S}_{ij}$: the first two contain the "horizontal" and "vertical" axes of the surface patch, and the third one is the homogeneous coordinate vector of its center.

To extract the matrices $\mathcal{N}_j$ (and thus the corresponding patches' geometry) from a set of image measurements, we construct a reduced factorization of $\hat{\mathcal{S}}$ by picking, as in [39], the center of mass of the surface patches' centers as the origin of the world coordinate system, and the center of mass of these points'

projections as the origin in each image. In this case, the projection equation $\mathcal{S}_{ij} = \mathcal{M}_i \mathcal{N}_j$ becomes

$$\begin{bmatrix} \mathcal{D}_{ij} \\ 0\ 0\ 1 \end{bmatrix} = \begin{bmatrix} \mathcal{A}_i & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathcal{B}_j \\ 0\ 0\ 1 \end{bmatrix}, \quad \text{or} \quad \mathcal{D}_{ij} = \mathcal{A}_i \mathcal{B}_j,$$

where $\mathcal{A}_i$ is a 2×3 matrix, $\mathcal{D}_{ij} = [\boldsymbol{h}\ \boldsymbol{v}\ \boldsymbol{c}]_{ij}$ is a 2×3 matrix, and $\mathcal{B}_j = [\boldsymbol{H}\ \boldsymbol{V}\ \boldsymbol{C}]_j$ is a 3×3 matrix. It follows that the reduced $2m \times 3n$ matrix

$$\hat{\mathcal{D}} = \hat{\mathcal{A}}\hat{\mathcal{B}}, \text{ where } \hat{\mathcal{D}} \stackrel{\text{def}}{=} \begin{bmatrix} \mathcal{D}_{11} & \dots & \mathcal{D}_{1n} \\ \vdots & \ddots & \vdots \\ \mathcal{D}_{m1} & \dots & \mathcal{D}_{mn} \end{bmatrix}, \ \hat{\mathcal{A}} \stackrel{\text{def}}{=} \begin{bmatrix} \mathcal{A}_1 \\ \vdots \\ \mathcal{A}_m \end{bmatrix}, \ \hat{\mathcal{B}} \stackrel{\text{def}}{=} [\, \mathcal{B}_1 \quad \dots \quad \mathcal{B}_n \,], \tag{1}$$

has at most rank 3. Following [39] we use singular value decomposition to factorize $\hat{\mathcal{D}}$ and compute estimates of the matrices $\hat{\mathcal{A}}$ and $\hat{\mathcal{B}}$ that minimize the squared Frobenius norm of the matrix $\hat{\mathcal{D}} - \hat{\mathcal{A}}\hat{\mathcal{B}}$. Geometrically, the (normalized) Frobenius norm $d = |\hat{\mathcal{D}} - \hat{\mathcal{A}}\hat{\mathcal{B}}|/\sqrt{3mn}$ of the residual can be interpreted as the root-mean-squared reprojection error, that is, the distance (in pixels) between the center and side points of the patches observed in the image and those predicted from the recovered matrices $\hat{\mathcal{A}}$ and $\hat{\mathcal{B}}$. Given n matches established across m images (a match is an m-tuple of image patches), the residual error d can thus be used as a measure of inconsistency between the matches.

2.3 Matching

Matching is a fundamental process in both modeling and recognition. An image can be viewed as simply a collection of 2D patches, and likewise a 3D model is a collection of 3D patches. There are three steps in our general procedure for matching between two such patch sets A and B:

Step 1 — Appearance based selection of potential matches. For each patch in set A, this step selects one or more patches in set B with similar appearance, as measured by the descriptors presented in Section 2.1. Mismatches might occur due to measurement noise or confusion of similar (for example, repetitive) structures.

Step 2 – Robust estimation. Using RANSAC, alignment, or other related techniques, this step selects a geometrically consistent subset of the match hypotheses. Our assumption is that the largest such consistent set will contain mostly true matches. This establishes the geometric relationship between the two sets of patches A and B.

Step 3 – Geometry-based addition of matches. This step seeks a fixed-point in the space $(A \times B)$ of matches by iteratively estimating a geometric model based on the current set of matches and then selecting all match hypotheses that are consistent with the model. At the same time it adds new match hypotheses

guided by the model. Generally, the geometric model will not change much during this process. Rather, the resulting maximal set of matches benefits recognition, where the number of matches acts as a confidence measure, and modeling, where it produces better coverage of the object.

3 3D Object Modeling from Images

There are several combinatorial and geometric problems to solve in order to convert a set of images into a 3D model. The overall process is divided into four steps: (1) *matching*: match regions between pairs of images; (2) *chaining*: link matches across multiple images; (3) *stitching*: solve for the affine structure and motion while coping with missing data; (4) *Euclidean upgrade*: use constraints associated with the intrinsic parameters of the camera to turn the affine reconstruction into a Euclidean one. In the following we describe each of these steps. We will use a teddy bear to illustrate some of the steps of the modeling process. Additional modeling experiments will also be presented.

Matching. The first step is to match the regions found in a pair of images. This is an instance of the *wide-baseline stereo matching* problem which has been well studied in the literature [3, 22, 24, 31, 35, 38, 44]. Any technique that generates a set of matches between affine regions in a pair of images is appropriate, including the general matching procedure (Section 2.3). This algorithm appears in three different contexts in this work, so we have chosen to give the details of its application only in the object recognition case (Section 4). Here we give a very brief sketch of its application to 2D matching. For the appearance-based matching (Step 1) we compare SIFT descriptors. For robust estimation (Step 2) we take advantage of the normalized residual $d = |\hat{\mathcal{D}} - \hat{\mathcal{A}}\hat{\mathcal{B}}|/\sqrt{3mn}$ to measure the consistency of subsets of the matches. Finally, in Step 3 we use an estimate of the epipolar geometry between the two images to find additional hypothetical matches, which are again filtered using the consistency measure. For details on the 2D matching procedure, see [33].

Chaining. The matching process described in the previous section outputs affine regions matched across pairs of views. It is convenient to represent these matches by a single (sparse) *patch-view* matrix whose columns represent surface patches, and rows represent the images in which they appear (Figure 5).

There are two challenges to overcome in the chaining process. One is to ensure that the image measurements $\mathcal{S}_{ij}$ are self-consistent for all projections of a given patch j. To solve this, we choose one member of the corresponding column as reference patch, and refine the parameters of the other patches to maximize their texture correlation with it (Figure 6). The second challenge is to cope with mismatches, which can cause two patches in one image to be associated with the same column in the patch-view matrix. In order to properly construct the matrix, we choose the one patch in the image whose texture is closest to the reference patch mentioned above.

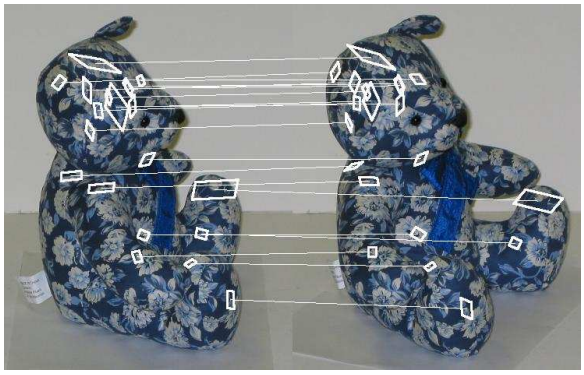


Fig. 4. Some of the matches found in two images of the bear (for readability, only 20 out of hundreds of matches are shown here). Note that the lines drawn in this diagram are *not* epipolar lines. Instead they indicate pairs of matched affine regions.

Fig. 5. A (subsampled) patch-view matrix for the teddy bear. The full patch-view matrix has 4,212 columns. Each black square indicates the presence of a given patch in a given image.

Stitching. The patch-view matrix is comparable to the data matrix used in factorization approaches to affine structure from motion [39]. If all patches appeared in all views, we could indeed factorize the matrix directly to recover the patches' 3D configurations as well as the camera positions. In general, however, the matrix is sparse. To cope with this, we find dense blocks (sub-matrices with complete data) to factorize and then register ("stitch") the resulting sub-models into a global one. The problem of finding maximal dense blocks within the patch-view matrix reduces to the NP-complete problem of finding maximal cliques in a graph. In our implementation, we use a simple heuristic strategy which, while not guaranteed to be optimal or complete, generally produces an adequate solution: Briefly, we find a dense block for each patch—that is, for each column in the patch-view matrix—by searching for all other patches that are visible in at least the same views. In practice, this strategy provides both a good coverage of the data by dense blocks and an adequate overlap between blocks.

The factorization technique described in Section 2.2 can of course be applied to each dense block to estimate the corresponding projection matrices and patch configurations in some local affine coordinate system. The next step is to combine the individual reconstructions into a coherent global model, or equivalently

Fig. 6. Refining patch parameters across multiple views: rectified patches associated with a match in four views before (top) and after (bottom) applying the refinement process. The patch in the rightmost column is used as a reference for the other three patches. The errors shown in the top row are exaggerated for the sake of illustration.

register them in a single coordinate system. With a proper set of constraints on the affine registration parameters, this can easily be expressed as an eigenvalue problem. In our experiments, however, we have found this linear approach to be numerically ill behaved (this is related to the inherent affine *gauge ambiguity* of our problem). Thus, in practice, we pick an arbitrary block as *root*, and iteratively register all others with this one using linear least squares, before using a non-linear *bundle adjustment* method to refine the global registration parameters.

Euclidean Upgrade. It is not possible to go from affine to Euclidean structure and motion from two views only [14]. When three or more views are available, on the other hand, it is a simple matter to compute the corresponding Euclidean weak-perspective projection matrices (assuming zero skew and known aspect ratios) and recover the Euclidean structure [39, 30]: Briefly, we find the 3×3 matrix $\mathcal{Q}$ such that $\mathcal{A}_i\mathcal{Q}$ is part of a scaled rotation matrix for $i = 1, \ldots, m$. This provides linear constraints on $\mathcal{Q}\mathcal{Q}^T$, and allows the estimation of this symmetric matrix via linear least-squares. The matrix $\mathcal{Q}$ can then be computed via Cholesky decomposition [29, 45].

Modeling results. Figure 7 shows a complete model of the teddy bear, along with the directions of the affine cameras. Figure 8 shows the models (but not the cameras) for seven other objects. The current implementation of our modeling approach is quite reliable, but rather slow: The teddy bear shown in Figure 7 is our largest model, with 4014 model patches computed from 20 images (24 image pairs). Image matching takes about 75 minutes per pair using the general matching procedure (Section 2.3), for a total of 29.9 hours. (All computing times

in this presentation are given for C++ programs executed on a 3Ghz Pentium 4 running Linux.) The remaining steps to assemble the model run in 1.5 hours. The greatest single expense in our modeling procedure is patch refinement, and this can be sped up by loosening convergence criteria and reducing the number of pixels processed, at the cost of a small loss in the number of matches.

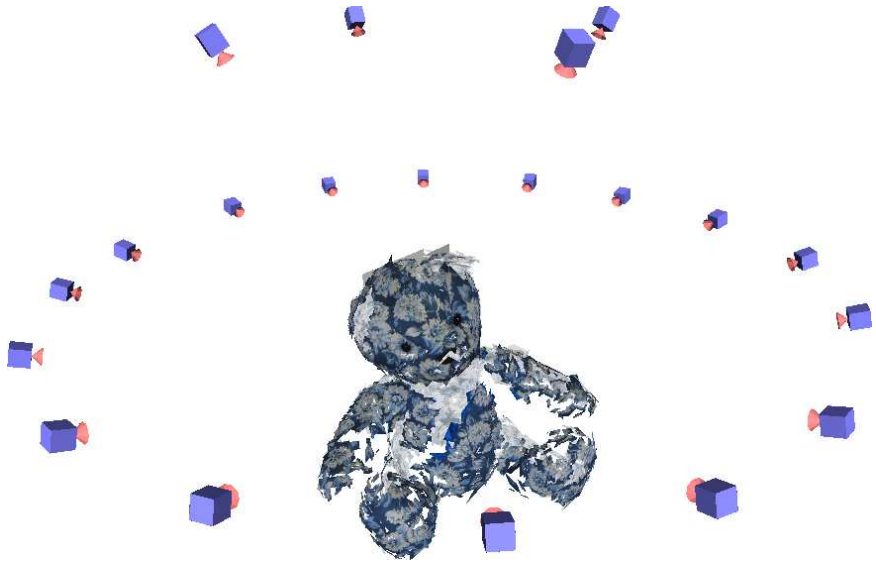
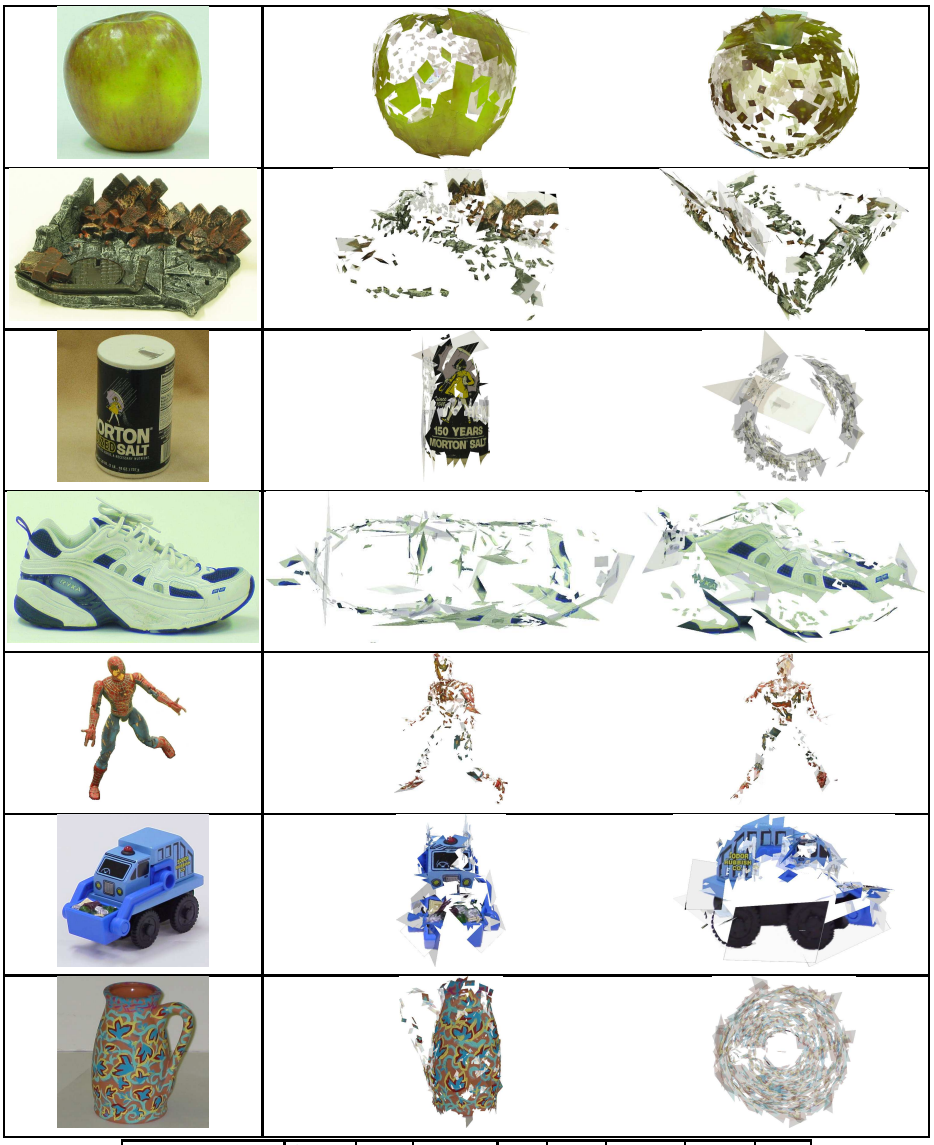


Fig. 7. The bear model, along with the recovered affine viewing directions. These cameras are shown at an arbitrary constant distance from the origin.

4 3D Object Recognition

We now address the problem of identifying instances of 3D models in a test image. This is essentially a matching process, and we apply again the general matching procedure (Section 2.3). The rest of this section describes the specifics of each step of the procedure.

Step 1 – Appearance based selection of potential matches. When texture patches have high contrast (that is, high variance in the intensity gradient) the SIFT descriptor does a good job of selecting promising matches. When the patches have low contrast SIFT becomes less reliable, since the intensity gradient field forms the basis for both the characteristic orientation and the histogram entries. In some situations, SIFT will even place the correct match in the bottom half of the list of candidates (Figure 9). For better reliability, we pre-filter the matches using a color descriptor: a 10×10 histogram of the UV portion of YUV space.



	Apple	Bear	Rubble	Salt	Shoe	Spidey	Truck	Vase
Input images	29	20	16	16	16	16	16	20
Model patches	759	4014	737	866	488	526	518	1085

Fig. 8. Object gallery. Left column: One of several input pictures for each object. Right column: Renderings of each model, not necessarily in the same pose as the input picture. Top to bottom: An apple, rubble (Spiderman base), a salt can, a shoe, Spidey, a toy truck, and a vase.

We compare the color descriptors using χ^2 distance and eliminate those below a threshold. Unfortunately, color is also unreliable due to variation in the spectral content of light sources and in the spectral response of sensors. Therefore we use a contrast measure to guide the choice between tight and loose thresholds in the color filtering step. This effectively shifts credence between the color and SIFT descriptors on an individual patch basis.

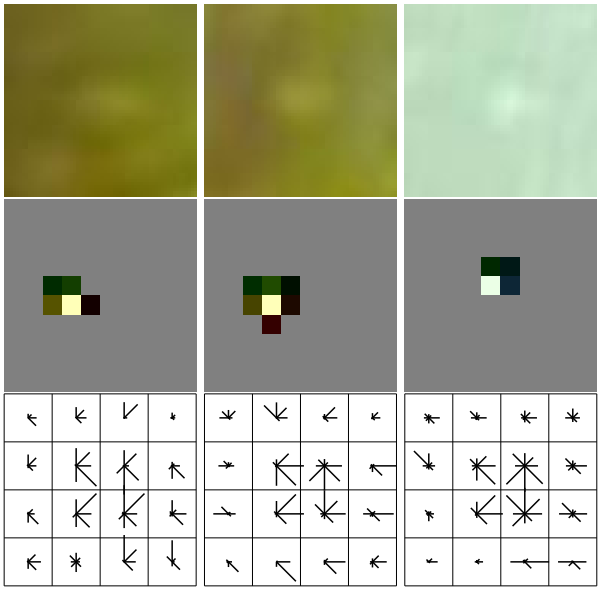


Fig. 9. Comparing SIFT and color descriptors on low-contrast patches. The center column is the model patch. The left column is the correct match in the image. The right column is the match in the image ranked first by SIFT (but that is in fact an incorrect match). The top row shows the patch, the middle row shows the color histogram, and the bottom row shows the SIFT descriptor. The incorrect match has a Euclidean distance of 0.52 between SIFT descriptors and a χ^2 distance of 1.99 between the corresponding color histograms; and the correct match has a SIFT distance of 0.67 and a color distance of 0.03. The two patches on the left are red and green, while the patch on the right is aqua.

Step 2 – Robust Estimation. This step finds the largest geometrically consistent set of matches. First, we apply neighborhood constraints to discard obviously inconsistent matches (Figure 10): For each match we construct the projection matrix (since a Euclidean model is available and a match contains three points) and use it to project the surrounding patches. If they lie close, the match is kept. Second, we refine the matched image regions with non-linear least squares to maximize their correlation with the corresponding model patches. This is the most expensive step, so we apply it after the neighborhood constraint.

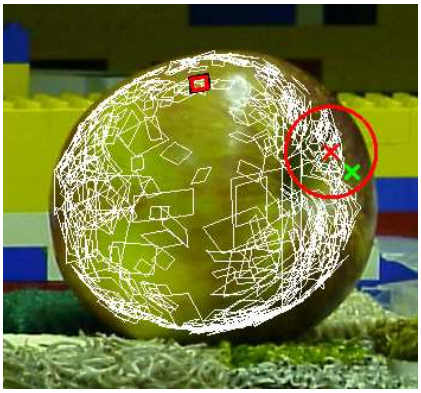


Fig. 10. An illustration of the neighborhood constraint. The small parallelogram in the upper center is the one used to estimate the projection matrix. The white parallelograms are projections of other forward-facing patches in the 3D model. The "×" surrounded by a circle is the center of one of the patches being tested, and the other "×" within the circle is its match in the image.

Various methods for finding matching features consistent with a given set of geometric constraints have been proposed in the past, including interpretation tree (or alignment) techniques [2, 6, 11, 13, 19], geometric hashing [15, 16], and robust statistical methods such as RANSAC [8] and its variants [40]. Both alignment and RANSAC can easily be implemented in the context of the general matching procedure (Section 2.3). We used several alternatives in our experiments, and found that the following "greedy" variant performed best: Let M be the number of matches found by appearance (typically limited to 12,000). For each match, we construct a "seed" model by iteratively adding the next most compatible match, just as in alignment, until the total matches in the seed reach a limit N (typically set to 20). Then we use the model constructed from this seed to collect a consensus set, just as in RANSAC. Thus, the "greedy" variant is a hybrid between alignment and RANSAC.

Step 3 – Geometry-Based Addition of Matches. The matches found by the estimation step provide a projection matrix that places the model into the image. All forward-facing patches in the model could potentially be present in the image. Therefore, we project each such model patch and select the K (typically 5 or 10) closest image patches as new match hypotheses.

Object Detection. Once an object model has been matched to an image, some criterion is needed to decide whether it is present or not. We use the following one:

(number of matches $\geq m$ OR matched area/total area $\geq a$) AND distortion $\leq d$,

where nominal values for the parameters are $m = 10$, $a = 0.1$, and $d = 0.15$. Here, the measure of distortion is

$$\frac{\boldsymbol{a}_1^T \boldsymbol{a}_2}{|\boldsymbol{a}_1||\boldsymbol{a}_2|} + \left(1 - \frac{\min(|\boldsymbol{a}_1|, |\boldsymbol{a}_2|)}{\max(|\boldsymbol{a}_1|, |\boldsymbol{a}_2|)}\right),$$

where $\boldsymbol{a}_i^T$ is the ith row of the leftmost 2×3 portion $\mathcal{A}$ of the projection matrix, and it reflects how close this matrix is to the top part of a scaled rotation matrix. The matched surface area of the model is measured in terms of the patches whose normalized correlation is above the usual thresholds, and it is compared to the total surface area actually visible from the predicted viewpoint.

Recognition results. Our recognition experiments match all eight of our object models against a set of 51 images. Each image contains instances of up to five object models, though the typical image only contains one or two. Using the nominal values for the detection parameters given above, the method gives no false positives and a recognition rate (averaged over the eight object models) of 94%.

Figure 11 shows a comparison study including our method and several other state-of-the-art object recognition systems. Our dataset is publicly available at `http://www-cvr.ai.uiuc.edu/ponce_grp/data`, and several other research groups graciously provided test results on it using their systems. The specific algorithms tested were the ones proposed by Ferrari, Tuytelaars & Van Gool [7], Lowe [20], Mahamud & Hebert [21], and Moreels, Maire & Perona [25]. In addition, we performed a test using our wide-baseline matching procedure between a database of training images and the test set, without using 3D models. For details of the comparative study, see [33].

Figure 12 shows sample results of some challenging (yet successful) recognition experiments, with a large degree of occlusion and clutter. Figure 13 shows the images where recognition fails. Note the views where the shoe fails. These are separated by about 60° from the views used during modeling. The surface of the shoe has a very sparse texture, so it is difficult to reconstruct some of the shape details. These details become more significant when the viewpoint moves from nearly parallel to the surface normal to nearly perpendicular.

5 Video

Modeling from video (contiguous image sequences) is similar in many respects to modeling from still images. In particular, we can use the same methods for describing the appearance and the geometric structure of affine-covariant patches. Establishing correspondence between multiple views of the same patch is actually easier in video sequences, since successive frames are close to each other in space and time, and it is sufficient to use tracking rather than wide-baseline matching. On the other hand, the problem of modeling from video is made *much* more difficult by the presence of multiple independently moving objects. To cope

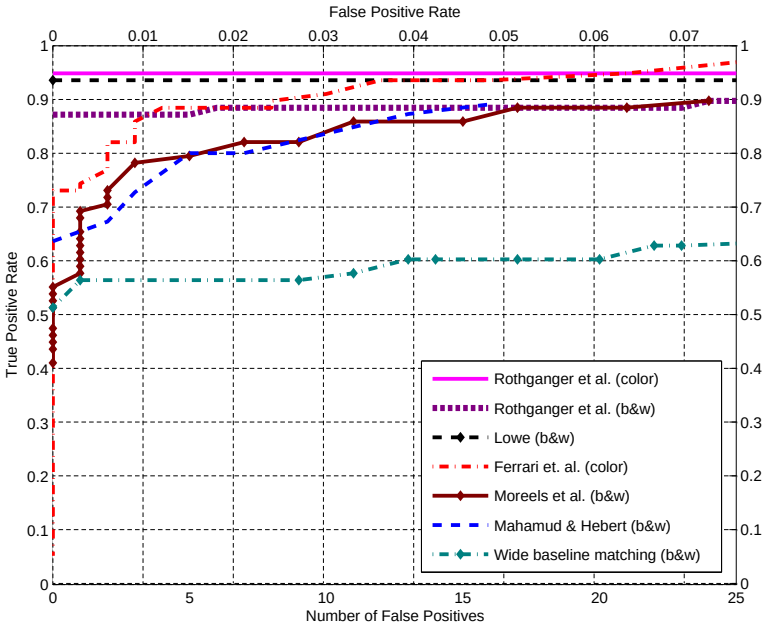


Fig. 11. True positive rate plotted against number of false positives for several different recognition methods.

with this, we take advantage of the factorization and error measure presented in Section 2.2 to simultaneously segment the moving components and build their 3D models. The resulting piecewise-rigid 3D models can be directly compared using the general matching procedure (Section 2.3), promising a method for video shot matching [1, 34, 37, 46].

The modeling process for video starts by extracting affine regions from the first frame and tracking them through subsequent frames. It continues to add new affine regions in each subsequent frame as old ones move out of view or die off for various reasons. The collection of all the tracked patches again forms a patch-view matrix. This matrix will in general contain more than one rigid component. Each rigid component has a different motion, producing a different set of projection matrices. If we attempt to construct a 3D patch for a track (column) using a set of cameras from a different rigid component, the reprojection error will be high, while constructing a 3D patch using cameras from the same rigid component will produce a low error. This fact leads to a motion segmentation technique based on RANSAC [9, 41]. The basic procedure is to locate a section of the video with a large number of overlapping tracks (that is, a large number of visible patches), select a random pair of them to reconstruct a set of cameras, and then construct a consensus set by measuring the reprojection error associated with each of the remaining tracks and adding those below a threshold. The largest consensus set becomes the basis of a new rigid component. The new

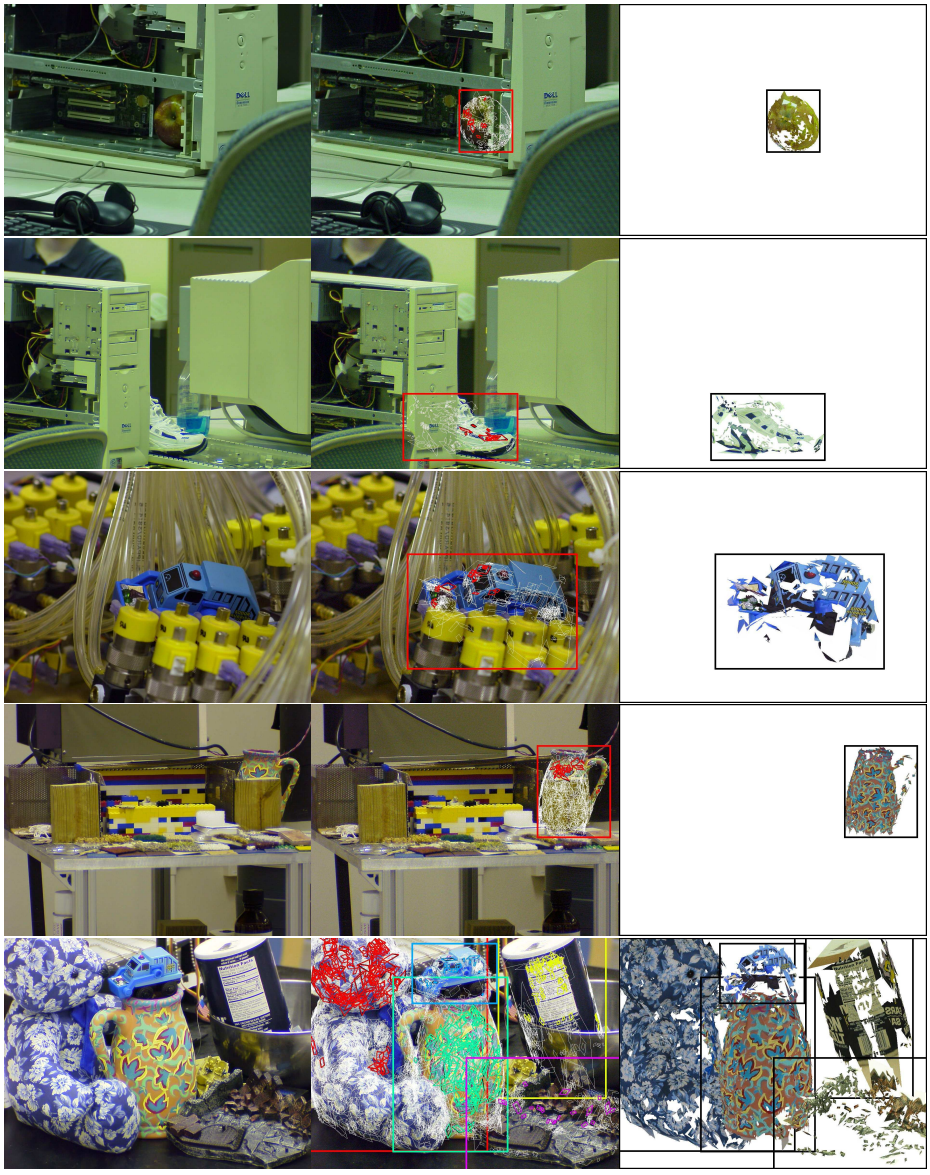


Fig. 12. Some challenging but successful recognition results. The recognized models are rendered in the poses estimated by our program, and bounding boxes for the reprojections are shown as rectangles.

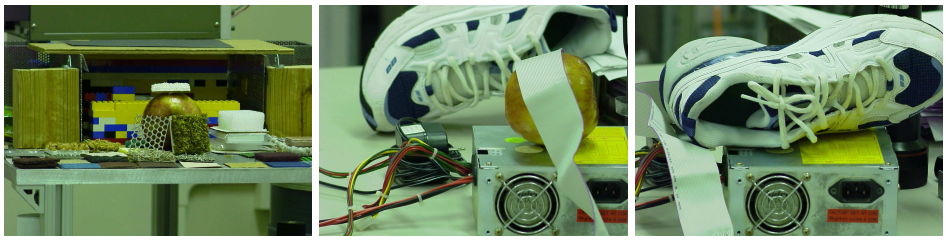


Fig. 13. Images where recognition fails.

model is propagated forward and backward through time, adding all compatible tracks. Finally, we remove the entire set of tracks, and repeat the procedure until all components of reasonable size have been found.

Rigid motion consistency may not be measured directly if two patches are not visible at the same time in the video. It is therefore necessary to extend the range of frames in the video covered by the working model as more consistent patches are found. The stitching method described in Section 3, while very accurate, is too expensive and not suited for building a model incrementally. Instead, we use a method called "bilinear incremental SFM" to add sparse measurements from the patch-view matrix to an existing model. Essentially, the method adds one row or column at a time from the patch-view matrix to a model, reconstructing one camera or patch respectively. It reconstructs patches using known cameras associated with the sparse set of image measurements in the new column, and similarly it reconstructs cameras using known patches associated with the image measurements in a row. At each step it always selects the next row or column that has the most image measurements overlapping the current model. In order to propagate the effects of new data, it periodically re-estimates all the cameras and patches currently in the model, exactly as in the resection-intersection method of bundle adjustment [42].

Experimental results. Figure 14 shows results of segmenting and modeling shots from the movies "Run Lola Run" and "Groundhog Day". These movies contain significant perspective effects, so we have used a more general projection model that is beyond the scope of this chapter, see [32] for details. The first row of the figure shows a scene from "Run Lola Run" where a train passes overhead. The detected components are the train and the background. The second row shows a corner scene from the same movie. The two rigid components are the car and the background. The third row of Figure 14 shows a scene from "Groundhog Day". The rigid components are the van and the background. Later, another vehicle turns off the highway and is also found as a component. The last row of the figure is a reprojection of the 3D model of the van. Note that the viewpoint of the reprojection is significantly different than any in the original scene.

Figure 15 shows the results of a recognition test over a set of 27 video shots collected from various sources: the movies "Run, Lola, Run" and "Groundhog Day", as well as several videos taken in the laboratory. Each scene appeared in

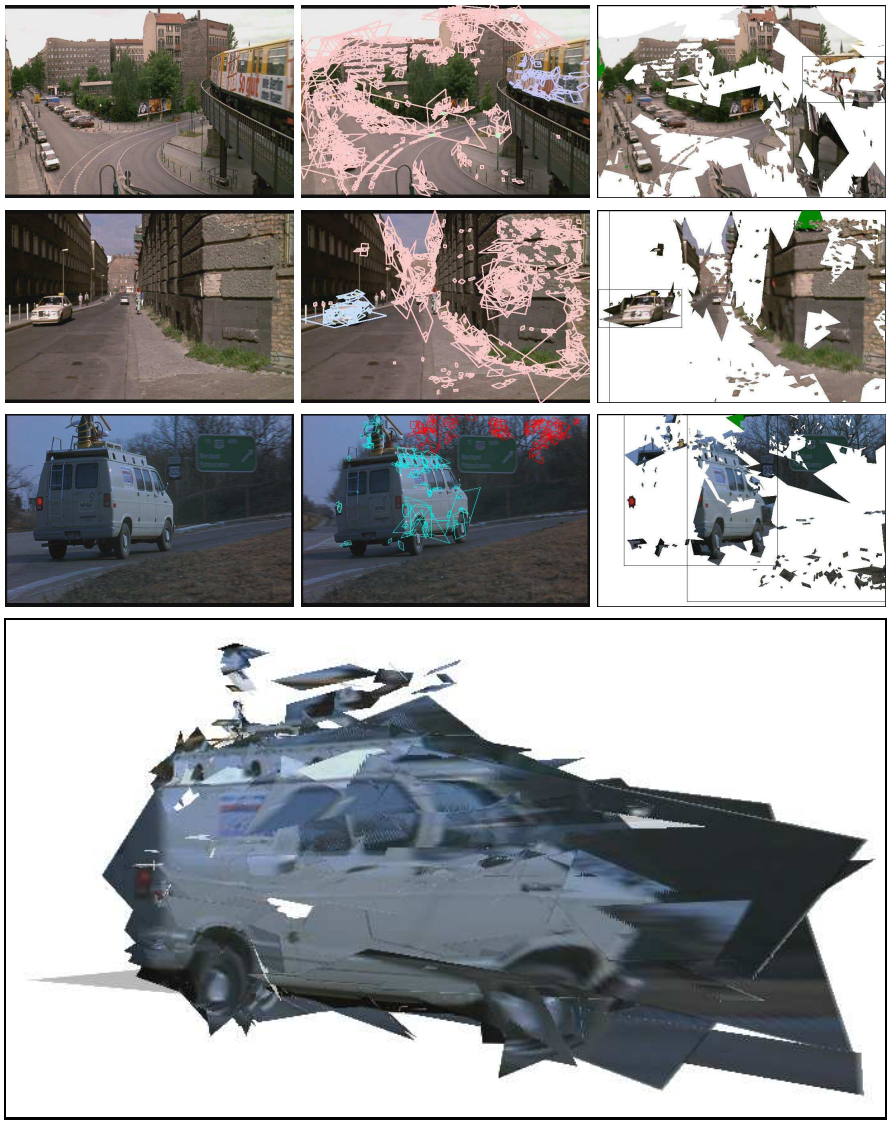


Fig. 14. Segmentation and modeling of shots from “Run Lola Run” and “Groundhog Day”.

2 or 3 of the shots. We selected 10 different 3D components in turn to act as queries, and used the general matching procedure (Section 2.3) between each query model and the rest of the set, see [32] for details.

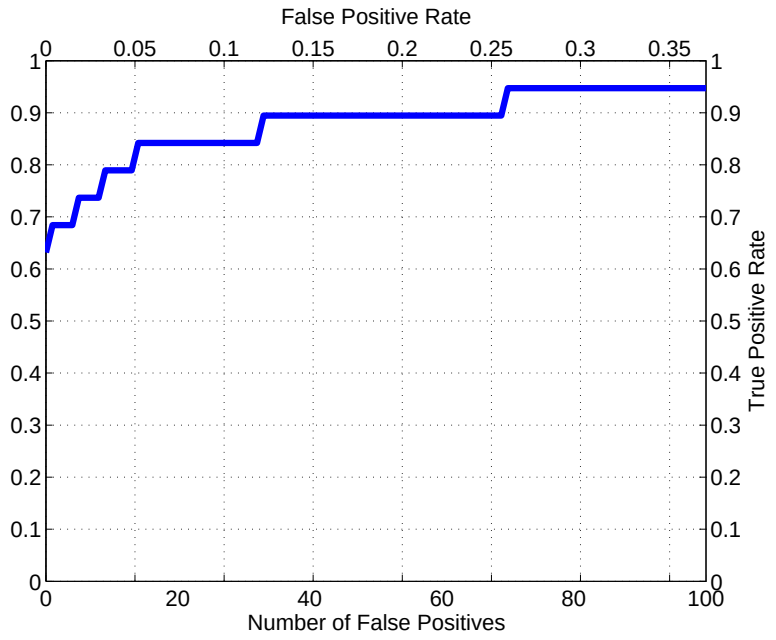


Fig. 15. Recognition rate versus false positives for a shot-matching test.

Figure 16 shows some of the correctly matched models. It shows a video frame from the recognized shot and a projection of the 3D model of the query shot. This demonstrates how well the two models are registered in 3D. These results are best viewed in motion, and sample videos appear on our web site: `http://www-cvr.ai.uiuc.edu/ponce_grp/research/3d`.

6 Discussion

We have proposed in this article to revisit invariants as a local object description that exploits the fact that smooth surfaces are always planar in the small. Combining this idea with the affine regions of Mikolajczyk and Schmid [24] has allowed us to construct a normalized representation of local surface appearance that can be used to select promising matches in 3D object modeling and recognition tasks. We have used multi-view geometric constraints to represent the larger 3D surface structure, retain groups of consistent matches, and reject incorrect ones. Our experiments demonstrate the promise of the proposed approach to 3D object recognition.

Fig. 16. Some correctly matched shots. The left image is the original frame of the test shot. The right image shows the query model reprojected into the test video.

We have extended our approach to automatically perform simultaneous motion segmentation and 3D modeling in video sequences containing multiple independently moving objects. Multi-view geometric constraints guide the selection of patches that move together rigidly and again represent their 3D surface structure, resulting in a set of rigid 3D components.

We have reduced 2D images, 3D models, image sequences and video scenes to a simple representation: a collection of affine patches. Any such collection may be matched to any other, aided by a representation of the geometric relationship between the two. We have presented three examples of such matching: between a pair of images (wide-baseline matching), between a 3D model and an image (object recognition), and between two 3D models (shot matching). In all cases, we first select match hypotheses based on appearance similarity and then find a subset that are geometrically consistent; and finally expand this set guided by both geometry and appearance.

Let us close by sketching several directions for improvement of the existing method. One such direction is increasing the computational efficiency of our current implementation. Two key changes would be to use a voting or indexing scheme rather than naive all-to-all matching, and to avoid patch refinement by developing more robustness to noise in the image measurements. Next, we plan to pursue various improvements to the feature extraction method. The current scheme depends in large part on corner-like Harris interest points, which often fall across object boundaries, and therefore cannot be matched or tracked reliably. To help overcome this problem, we could use maximally stable extremal regions [22], which tend to be detected on relatively "flat" regions of an object's surface. More generally, some 3D objects, such as bicycles and lamp-posts, are not amenable to representation by planar patches at all. In such cases, a hybrid system that models point, edge, and planar features would be more suitable. Finally, many interesting objects are non-rigid, the prime example being human actors. Thus, an important future research direction is extending our approach to deal with non-rigid, articulated objects.

Acknowledgments. This research was partially supported by the National Science Foundation under grants IIS-0308087 and IIS-0312438, Toyota Motor Cor-

poration, the UIUC-CNRS Research Collaboration Agreement, the European FET-open project VIBES, the UIUC Campus Research Board, and the Beckman Institute.

References

1. A. Aner and J. R. Kender. Video summaries through mosaic-based shot and scene clustering. In *European Conference on Computer Vision*, pages 388–402, Copenhagen, Denmark, 2002.
2. N. Ayache and O. D. Faugeras. Hyper: a new approach for the recognition and positioning of two-dimensional objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):44–54, January 1986.
3. A. Baumberg. Reliable feature matching across widely separated views. In *Conference on Computer Vision and Pattern Recognition*, pages 774–781, 2000.
4. J. B. Burns, R. S. Weiss, and E. M. Riseman. View variation of point-set and line-segment features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(1):51–68, January 1993.
5. O. D. Faugeras, Q. T. Luong, and T. Papadopoulo. *The Geometry of Multiple Images*. MIT Press, 2001.
6. O. D. Faugeras and M. Hebert. The representation, recognition, and locating of 3-D objects. *International Journal of Robotics Research*, 5(3):27–52, Fall 1986. 1986.
7. V. Ferrari, T. Tuytelaars, and L. Van Gool. Simultaneous object recognition and segmentation by image exploration. In *European Conference on Computer Vision*, 2004.
8. M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Communications ACM*, 24(6):381–395, June 1981.
9. A. W. Fitzgibbon and A. Zisserman. Multibody structure and motion: 3-d reconstruction of independently moving objects. In *European Conference on Computer Vision*, pages 891–906. Springer-Verlag, June 2000.
10. J. Gårding and T. Lindeberg. Direct computation of shape cues using scale-adapted spatial derivative operators. *International Journal of Computer Vision*, 17(2):163–191, 1996.
11. W. E. L. Grimson and T. Lozano-Pérez. Localizing overlapping parts by searching the interpretation tree. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(4):469–482, 1987.
12. C. Harris and M. Stephens. A combined edge and corner detector. In 4^{th} *Alvey Vision Conference*, pages 189–192, Manchester, UK, 1988.
13. D. P. Huttenlocher and S. Ullman. Object recognition using alignment. In *International Conference on Computer Vision*, pages 102–111, 1987.
14. J. J. Koenderink and A. J. van Doorn. Affine structure from motion. *Journal of the Optical Society of America*, 8(2):377–385, February 1991.
15. Y. Lamdan and H. J. Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. In *International Conference on Computer Vision*, pages 238–249, 1988.
16. Y. Lamdan and H. J. Wolfson. On the error analysis of 'geometric hashing'. In *Conference on Computer Vision and Pattern Recognition*, pages 22–27, Maui, Hawaii, 1991.

17. T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):77–116, 1998.
18. T. Lindeberg and J. Gårding. Shape-adapted smoothing in estimation of 3-D depth cues from affine distortions of local 2-D brightness structure. In *European Conference on Computer Vision*, pages 389–400, Stockholm, Sweden, May 2-5 1994. Springer-Verlag Lecture Notes in Computer Science, vol. 800.
19. D. G. Lowe. The viewpoint consistency constraint. *International Journal of Computer Vision*, 1(1):57–72, 1987.
20. D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
21. S. Mahamud and M. Hebert. The optimal distance measure for object detection. In *Conference on Computer Vision and Pattern Recognition*, 2003.
22. J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *British Machine Vision Conference*, volume I, pages 384–393, 2002.
23. K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *International Conference on Computer Vision*, pages 525–531, Vancouver, Canada, July 2001.
24. K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *European Conference on Computer Vision*, volume I, pages 128–142, 2002.
25. P. Moreels, M. Maire, and P. Perona. Recognition by probabilistic hypothesis construction. In *European Conference on Computer Vision*, 2004.
26. J. L. Mundy and A. Zisserman. *Geometric Invariance in Computer Vision*. MIT Press, 1992.
27. J. L. Mundy, A. Zisserman, and D. Forsyth. *Applications of Invariance in Computer Vision*, volume 825 of *Lecture Notes in Computer Science*. Springer-Verlag, 1994.
28. H. Murase and S. K. Nayar. Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision*, 14:5–24, 1995.
29. C. J. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):206–218, 1997.
30. J. Ponce. On computing metric upgrades of projective reconstructions under the rectangular pixel assumption. In *Second SMILE Workshop*, pages 18–27, 2000.
31. P. Pritchett and A. Zisserman. Wide baseline stereo matching. In *International Conference on Computer Vision*, pages 754–760, Bombay, India, 1998.
32. F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. Segmenting, modeling, and matching video clips containing multiple moving objects. In *Conference on Computer Vision and Pattern Recognition*, volume 2, pages 914–921, Washington, D.C., June 2004.
33. F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. 3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. *International Journal of Computer Vision*, 2005. To appear.
34. F. Schaffalitzky and A. Zisserman. Automated scene matching in movies. In *Proceedings of the Challenge of Image and Video Retrieval*, London, 2002.
35. F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or "how do i organize my holiday snaps?". In *European Conference on Computer Vision*, volume I, pages 414–431, 2002.
36. C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–535, May 1997.

37. J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *International Conference on Computer Vision*, 2003.
38. D. Tell and S. Carlsson. Wide baseline point matching using affine invariants computed from intensity profiles. In *Proc 6th ECCV*, pages 814–828, Dublin, Ireland, June 2000. Springer LNCS 1842-1843.
39. C. Tomasi and T. Kanade. Shape and motion from image streams: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.
40. P. Torr and A. Zisserman. Mlesac: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138–156, 2000.
41. P. Torr. *Motion Segmentation and Outlier Detection*. PhD thesis, University of Oxford, 1995.
42. B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment - a modern synthesis. In B. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms*, pages 298–372, Corfu, Greece, September 1999. Spinger-Verlag. LNCS 1883.
43. M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, Winter 1991.
44. T. Tuytelaars and L. Van Gool. Matching widely separated views based on affine invariant regions. *International Journal of Computer Vision*, 59(1):61–85, 2004.
45. D. Weinshall and C. Tomasi. Linear and incremental acquisition of invariant shape models from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):512–517, 1995.
46. M. M. Yeung and B. Liu. Efficient matching and clustering of video shots. In *International Conference on Image Processing*, volume 1, pages 338–341, Washington D.C., October 1995.